

## Inter-Office Memorandum

**DRAFT - DRAFT - DRAFT - DRAFT**

To	Cedar Interest Group	Date	February 17, 1981 10:51 AM
From	Jim Morris	Location	Palo Alto
Subject	Cedar Status	Organization	CSL

# XEROX

Filed on: [Ivy]<CedarDocs>Status.memo

This memo is a somewhat updated and elaborated version of the end of 1980 activity report for the Cedar project. The project is best viewed as three levels of activity: the *core* which consists of the microcode, operating system, and language, fundamental *tools* which consists of the editor, debugger, and system modeller, and *packages* most notable of which are the data base and graphics packages. The core system now runs on both Dolphin and Dorado processors. Several new language features, including run-time types, have appeared; and the performance of automatic storage management has been improved. Basic building blocks for the tools have been written and are in regular use, though not on Pilot. The tools themselves are still being designed. Packages for data base management, graphics, strings, lists and atoms are available and are in regular use.

### *Core*

Kernel: We have made modifications to the Pilot operating system and related utilities to enable them to execute on the Dorado processor. These involve changes to support a different input/output configuration, most notably, a different kind of rigid disk. There has been extensive work on Dorado and Dolphin microcode to support Pilot and various special input/output devices. [Taft, Levin, Fiala]

Cascade: We have constructed a system called Cascade, which represents the first stage in the transition to a Pilot-based program development environment. Cascade provides the facilities to edit, compile, bind, execute and debug Mesa programs under Pilot without resorting to Alto emulation. To reach this point, we converted the standard SDD Alto/Mesa compiler and binder to run in the CoPilot environment, and built an Executive and other utilities to control the program development cycle. Cascade is presently in use by Cedar implementers as well as the Mesa group in SDD. [Levin, Atkinson, Schmidt]

People who are currently using Pilot and Cascade are: Roy Levin, Russ Atkinson, Dan Swinehart, Eric Schmidt, Mark Brown, Jim Morris.

Language: We have continued to make progress on the conceptual foundations of the Abstract Machine. The aspects of the machine dealing with Mesa values have been implemented and are

**DRAFT - DRAFT - DRAFT - DRAFT**

being used to support the executive and debugging facilities discussed below. [Rovner]

We have improved the compiler, runtime, and Dorado microcode to make incremental collection run faster and semi-concurrently with its clients. We designed modifications that allow the collector to safely and efficiently trace the stacks of processes which deal with collectible storage. A trace-and-sweep garbage collector has been designed and an exploratory implementation has been tested. We wrote a facility for finalization of collectible objects. [Rovner, Atkinson, Satterthwaite, Haugeland]

New language features were implemented: variable sized arrays, better default initialization, support for dynamic types, LISP-like atoms, and lists. Progress was made on the Cedar/Mesa Primer and Reference manual. An exploratory remote procedure call facility was designed and built. A design for rapid turnaround for minor program changes was completed. [Rovner, Horning, McKeeman, Satterthwaite]

The new language features have been used by: Russ Atkinson, Ed Satterthwaite, Jim Morris, Warren Teitelman, Paul Rovner.

### *Tools*

Facilities for user terminal input have been written. The client sees an input stream of events such as keystrokes, mouse clicks and chords. The interface also provides a procedure for repositioning the stream as a function of time. This interface is being used by Warren Teitelman and Scott McGregor. [Swinehart]

Based on the graphics package provided by ISL, Document classes for file and stream documents have been written. As an exercise, a Document class for displaying LSI layouts was implemented. Scrolling and partial refresh of the screen are operational. A facility for attaching a stream, e.g. the keyboard stream, to a document has been provided. This has been used by Will Crowther. [Teitelman]

Based on input streams we wrote a programmable scanner that accepts a sequence of characters from a stream, and returns a dynamically typed Mesa value. An inverse function takes a value delivers a sequence of characters to a stream. Based upon these procedures and Documents, a user executive patterned after Interlisp's has been written. It accepts inputs of procedure names and argument values, applies the procedure to the arguments, and prints the result. Spelling correction and a REDO facility have been implemented. [Teitelman]

Some progress has been made on the design of a comprehensive set of debugging capabilities. A number of special Document types will be used to display understandable representations of the program sources and active data for an instance of an application. Operations available through menus related to these Documents will allow modification of data values and program control state. It will be possible to have one or more independent debugging environments for each simultaneously active application. [Swinehart]

System Models: We wrote programs that will recompile a system of Mesa modules, and will transport consistent versions of systems between machines. These systems operate on an as-needed

basis to avoid gratuitous compilation and file transfer. These programs are used by a large number of programmers. We built a database, using the Database package, that contains information about Mesa modules, their dependencies, and their whereabouts on remote file servers. This effort helped discover bugs in the Database package and Juniper.

The design of the system modelling language was refined. A program to convert from the existing descriptions configuration descriptions and other existing Mesa files to system models is largely complete. A program that takes these models and determines what needs to be compiled and transferred is partially implemented. [Lampson, Schmidt]

### *Packages*

Graphics: An extensive graphics management package has been implemented and is being used by Warren Teitleman. It is described more fully elsewhere. [J. Warnock, D. Wyatt]

Data Base Management: We completed a stable version of the system for use by applications. This includes an implementation of the DBTuples interface, plus utilities for database dump and reload. The first application in regular use is the CSL Notebook, a database holding CSL memoranda. The Notebook can be updated through the message system, and periodic summaries of new additions are distributed via the same medium. A second application is a database of information about Mesa modules; this supports system modelling activities. A prototype database browser, Palm, has been used to support personal databases. [M. Brown, R. Cattell, N. Suzuki].

Library: We instituted CedarLib, a facility for user-maintained modules, and distributed many new packages, including a widely-used random number package, IEEE floating point support, a general cache package, support for strings, and initial system modeling software. Software for automatically maintaining CedarLib abstracts and user lists was written. [Atkinson]